Eclipse Plug-in Development Research Project

Report 6

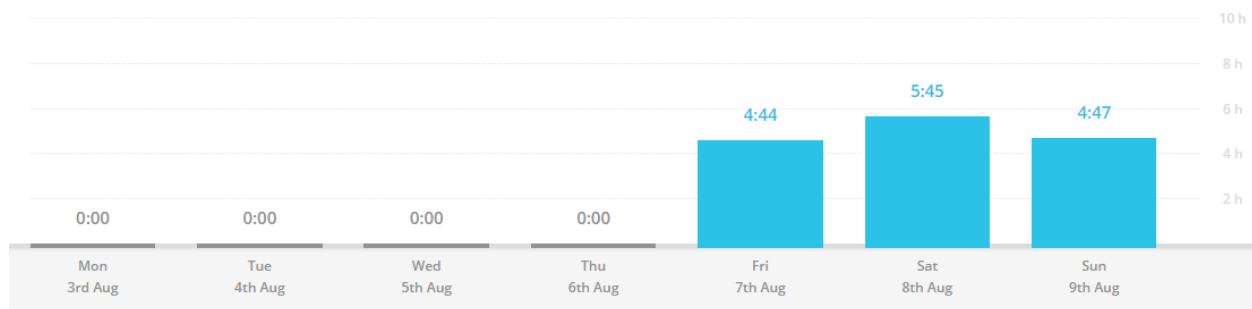UNIVERSITY AT BUFFALO

August 10, 2015
Authored by: Chern Yee Chua

Introduction

Due to the surgery of my knee, I had to take a week or two off to rest. In this week, I manage to get the comment block working but I am still getting no luck with the AST changes.

Synopsis

I spend roughly 15 hours on this project since last meeting. (I am shifting 10 hours to previous week: 7/19 – 7/25 on the time sheet)

0:00	0:00	0:00	0:00	4:44	5:45	4:47
Mon 3rd Aug	Tue 4th Aug	Wed 5th Aug	Thu 6th Aug	Fri 7th Aug	Sat 8th Aug	Sun 9th Aug

Eclipse Plugin	● 15:16:00
Detect comments in editor	4:47:00
Finding differences between ast changes	5:45:00
Working on recordModifications() and rewrite	4:44:00

Problem statements

After spending quite some time figuring out the method that Professor Alphonce showed me from last meeting, I still couldn't get the recordModifications() and rewrite() to work for this case. From what I have been testing on, the result is not the difference between two consecutive ASTs, but

rather a modification on the latter AST and by that saying is, it still gives the whole elements on editor when printing to system console which is not what we are intended to get.

So, I moved on to the next part. What I am trying to do here is to print statements to console without repeating the elements that I have already printed. Sadly, this is not functional yet.

```java
for(int i = 0; i < node.getBody().statements().size(); i++){

    if(addVariableInMethod.containsValue(
            node.getBody().statements().get(i))){
        System.out.println("Should be doing nothing");
    }

    else{
        addVariableInMethod.put(methodVariableCount + "",  node.getBody().statements().get(i));
        System.out.print(addVariableInMethod.get(methodVariableCount+ ""));
        methodVariableCount++;
    }

}
```

Basically, what the code above does is to check if the statement is already stored in the HashMap. If it is, then we should skip to next statement. If not, then we add the statement object into *addVariableInMethod* HashMap. What I am getting at is the *if* statement is never been accessed even though the *addVariableInMethod* has the statement. I even print out the result to console and both yields the same output.

So, if the method above is working, then we can get the newly added changes only instead of whole chunk of code inside method.

Note: the *methodVariableCount* is just a temporary name for the key-value pair in HashMap and we don't really care about it in this case.

Conclusion

I have learned SQLite recently and I was thinking maybe I could migrate from file output style to a local database scheme. Something like this:

ID	Name	Type	Category	Time	...
1	count	int	Field	2:59PM	
2	Com.example.sample		ImportDeclaration	3:03PM	
3	IssacClass	Class	Type	3:03PM	
4	IssacMethod	Method	Method	4:00PM	

...					

That is just a very rough sketch of how the database should look like. In the future, probably I might need to have a couple of tables for different purposes, such as a table just for elements inside a method, something like that and link them to primary table using foreign key and such.

As for the AST part, I will spend more time on that matter. It is just that I couldn't really get focused these days. We definitely could meet some time in this week again to catch up. Once again, sorry for the delay of work and inconvenience.

Thank you very much!